WHAT IS CLAIMED IS:

1   1.   A method for error detection in storage data, comprising:

2   retrieving one block from a plurality of blocks, wherein the retrieved block has a

3 pattern and a checksum field, and wherein the pattern is written into the plurality of

4 blocks during initialization of the blocks in a storage device;

5   determining whether a checksum computed from the retrieved block is different

6 than the value in the checksum field of the retrieved block;

7   determining whether the retrieved block includes the pattern; and

8   generating an error if the retrieved block does not include the pattern and if the

9 computed checksum is different than the value in the checksum field.

1   2.   The method of claim 1, wherein the pattern written into the plurality of

2 blocks while initializing data in the storage device is written during a format operation on

3 the storage device, wherein no checksum value is written to the checksum field during the

4 format operation.

1   3.   The method of claim 1, wherein the block is retrieved in response to a read

2 request from an application.

1   4.   The method of claim 1, further comprising:

2   writing data and a calculated checksum to one block after the pattern was written

3 to the blocks during initialization.

1   5.   The method of claim 4, wherein the checksum is calculated by:

2   computing the checksum from data received in a write request from an

3 application.

1   6.   The method of claim 1, wherein the checksum is computed with an

2   Exclusive-Or operation.

1   7.   The method of claim 1, wherein the pattern and checksum field are at

2   identical locations in each of the plurality of blocks.

1   8.   The method of claim 1, wherein the block of data retrieved from the

2   storage device is processed by a target driver, a device driver specific to the storage

3   device, and an adapter card specific to the storage device.

1   9.   The method of claim 8, wherein the block of data is retrieved by:

2   receiving at the target driver a first request for data from an application;

3   generating a second request corresponding to the first request by the target driver;

4   sending the second request from the target driver to the device driver;

5   generating a third request corresponding to the second request at the device driver,

6   wherein the third request is formatted to be compatible with the hardware of the storage

7   device; and

8   transmitting the third request from the device driver to the storage device via the

9   adapter card.

1   10.   The method of claim 1, wherein the first request is for data having a first

2   byte length and wherein the second request is for data having a second byte length longer

3   than the first byte length.

1   11.   A method for formatting a storage device into a plurality of blocks of a

2   block size, comprising:

3   reserving a set of bits within each of the plurality of blocks to store a pattern;

4   reserving a checksum field within each of the plurality of blocks to store a

5   checksum;

6   formatting the storage device into the plurality of blocks; and

7   during the formatting, writing the pattern into the set of bits within each of the

8   plurality of blocks, wherein no checksum value is written to the checksum field during

9   the formatting.

1   12.   The method of claim 11, where the checksum field occupies the same

2   location as the set of bits within each of the plurality of plurality of blocks.

1   13.   The method of claim 11, wherein the pattern and checksum field are at

2   identical locations in each of the plurality of blocks.

1   14.   The method of claim 11, wherein the block size on the storage device is

2   increased as a result of storing the checksum.

1   15.   The method of claim 11, wherein the formatting comprises executing a

2   SCSI FORMAT UNIT command, further comprising:

3   setting a FMTDATA field in the FORMAT UNIT command field within data

4   structures of the FORMAT UNIT command;

5   setting an IP field in a DEFECT LIST HEADER field within the data structures

6   of the FORMAT UNIT command;

7   setting a PATTERN TYPE to be hexadecimal 01 in an INITIALIZATION

8   PATTERN DESCRIPTOR field within the data structures of the FORMAT UNIT

9   command; and

10   setting an INITIALIZATION PATTERN LENGTH and an INITIALIZATION

11   PATTERN field in the INITIALIZATION PATTERN DESCRIPTOR field within the

12   data structures of the FORMAT UNIT command.

1   16.   The method of claim 15, wherin the SCSI FORMAT UNIT command is

2   configured by a host system and executed by the storage device.

1   17.   The method of claim 11, wherein the storage device is a member of a set

2   of storage devices comprising a Direct Access Storage Device (DASD), Just a Bunch of

3   Disks (JBOD), a Redundant Array of Independent Disks (RAID), a flexible disk, a rigid

4   disk, a tape library and an optical library.

1   18.   A system for error detection in storage data, wherein the storage data is

2   contained in a storage device, and wherein the storage data comprises a plurality of

3   blocks, the system comprising:

4   means for retrieving one block from the storage device, wherein the retrieved

5   block has a pattern and a checksum field, and wherein the pattern is written into the

6   plurality of blocks during initialization of the blocks in the storage device;

7   means for determining whether a checksum computed from the retrieved block is

8   different than the value in the checksum field of the retrieved block;

9   means for determining whether the retrieved block includes the pattern; and

10   means for generating an error if the retrieved block does not include the pattern

11   and if the computed checksum is different than the value in the checksum field.

1   19.   The system of claim 18, wherein the pattern written into the plurality of

2   blocks while initializing data in the storage device is written during a format operation on

3   the storage device, wherein no checksum value is written to the checksum field during the

4   format operation.

1   20.   The system of claim 18, further comprising:

2   means for writing data and a calculated checksum to one block after the pattern is

3   written to the blocks during initialization.

1   21.   The system of claim 18, further comprising processing the block retrieved

2   from the storage device with a target driver, a device driver specific to the storage device,

3   and an adapter card specific to the storage device.

1   22.   The system of claim 21, wherein the means for retrieving the block further

2   performs:

3   receiving at the target driver a first request for data from an application;

4   generating a second request corresponding to the first request by the target driver;

5   sending the second request from the target driver to the device driver;

6   generating a third request corresponding to the second request at the device driver

7   wherein the third request is formatted to be compatible with the hardware of the storage

8   device; and

9   transmitting the third request from the device driver to the storage device via the

10   adapter card.

1   23.   A system for formatting a storage device into a plurality of blocks of a

2   block size, comprising:

3   means for reserving a set of bits within each of the plurality of blocks to store a

4   pattern;

5   means for reserving a checksum field within each of the plurality of blocks to

6   store a checksum;

7   means for formatting the storage device into the plurality of blocks; and

8   during the formatting, means for writing the pattern into the set of bits within each

9   of the plurality of blocks, wherein no checksum value is written to the checksum field

10   during the formatting.

1   24.   The system of claim 23, wherein the means for formatting comprises

2   executing a SCSI FORMAT UNIT command, wherein the executing further comprises:

3   setting a FMTDATA field in the FORMAT UNIT command field within data

4   structures of the FORMAT UNIT command;

5   setting an IP field in a DEFECT LIST HEADER field within the data structures

6   of the FORMAT UNIT command;

7   setting a PATTERN TYPE to be hexadecimal 01 in an INITIALIZATION

8   PATTERN DESCRIPTOR field within the data structures of the FORMAT UNIT

9   command; and

10   setting an INITIALIZATION PATTERN LENGTH and an INITIALIZATION

11   PATTERN field in the INITIALIZATION PATTERN DESCRIPTOR field within the

12   data structures of the FORMAT UNIT command.

1   25.   A article of manufacture including code for error detection in storage data,

2   wherein the storage data is contained in a storage device, wherein the storage data

3   comprises a plurality of blocks, and wherein the code causes operations to be performed,

4   the operations comprising:

5   retrieving one block from the plurality of blocks, wherein the retrieved block has a

6   pattern and a checksum field, and wherein the pattern is written into the plurality of

7   blocks during initialization of the blocks in the storage device;

8   determining whether a checksum computed from the retrieved block is different

9   than the value in the checksum field of the retrieved block;

10   determining whether the retrieved block includes the pattern; and

11   generating an error if the retrieved block does not include the pattern and if the

12   computed checksum is different than the value in the checksum field.

1   26.   The article of manufacture of claim 25, wherein the pattern written into the

2   plurality of blocks while initializing data in the storage device is written during a format

3   operation on the storage device, wherein no checksum value is written to the checksum

4   field during the format operation.

1   27.   The article of manufacture of claim 25, wherein the block is retrieved in

2   response to a read request from an application.

1   28.   The article of manufacture of claim 25, further comprising:

2   writing data and a calculated checksum to one block after the pattern was written

3 to the blocks during initialization.

1   29.   The article of manufacture of claim 28, wherein the checksum is

2 calculated by:

3   computing the checksum from data received in a write request from an

4 application.

1   30.   The article of manufacture of claim 25, wherein the checksum is computed

2 with an Exclusive-Or operation.

1   31.   The article of manufacture of claim 25, wherein the pattern and checksum

2 field are at identical locations in each of the plurality of blocks.

1   32.   The article of manufacture of claim 25, wherein the block retrieved from

2 the storage device is processed by a target driver, a device driver specific to the storage

3 device, and an adapter card specific to the storage device.

1   33.   The article of manufacture of claim 32, wherein the block is retrieved by:

2   receiving at the target driver a first request for data from an application;

3   generating a second request corresponding to the first request by the target driver;

4   sending the second request from the target driver to the device driver;

5   generating a third request corresponding to the second request at the device driver

6 wherein the third request is formatted to be compatible with the hardware of the storage

7 device; and

8   transmitting the third request from the device driver to the storage device via the

9 adapter card.

1   34.   The article of manufacture of claim 25, wherein the first request is for data

2   having a first byte length and wherein the second request is for data having a second byte

1   length longer than the first byte length.

1   35.   An article of manufacture including code for formatting a storage device

2   into a plurality of blocks of a block size, wherein the code causes operations to be

3   performed, the operations comprising:

4   reserving a set of bits within each of the plurality of blocks to store a pattern;

5   reserving a checksum field within each of the plurality of blocks to store a

6   checksum;

7   formatting the storage device into the plurality of blocks; and

8   during the formatting, writing the pattern into the set of bits within each of the

9   plurality of blocks, wherein no checksum value is written to the checksum field during

10 the formatting.

1   36.   The article of manufacture of claim 35, where the checksum field occupies

2   the same location as the set of bits within each of the plurality of plurality of blocks.

1   37.   The article of manufacture of claim 35, wherein the pattern and checksum

2   field are at identical locations in each of the plurality of blocks.

1   38.   The article of manufacture of claim 35, wherein the block size on the

2   storage device is increased as a result of storing the checksum.

1   39.   The article of manufacture of claim 35, wherein the formatting comprises

2   executing of a SCSI FORMAT UNIT command, further comprising:

3   setting a FMTDATA field in the FORMAT UNIT command field within data

4   structures of the FORMAT UNIT command;

5   setting an IP field in a DEFECT LIST HEADER field within the data structures

6   of the FORMAT UNIT command;

7   setting a PATTERN TYPE to be hexadecimal 01 in an INITIALIZATION

8   PATTERN DESCRIPTOR field within the data structures of the FORMAT UNIT

9   command; and

10   setting an INITIALIZATION PATTERN LENGTH and an INITIALIZATION

11   PATTERN field in the INITIALIZATION PATTERN DESCRIPTOR field within the

12   data structures of the FORMAT UNIT command.

1   40.   The article of manufacture of claim 39, wherein the SCSI FORMAT UNIT

2   command is configured by a host system and executed by the storage device.

1   41.   The article of manufacture of claim 35, wherein the storage device is a

2   member of a set of storage devices comprising a Direct Access Storage Device (DASD),

3   Just a Bunch of Disks (JBOD), a Redundant Array of Independent Disks (RAID), a

4   flexible disk, a rigid disk, a tape library and an optical library.